



ANSIBLE

MODUL PEMBELAJARAN  
**TEKNOLOGI KOMPUTASI AWAN**  
ANSIBLE

## **Modul 2 Ansible**

**Dosen pengampu :**  
Henning Titi Ciptaningtyas, S.Kom, M.Kom.

**Disusun oleh :**  
Dimas Bagus Rachmadani  
I Putu Windy Arya Sagita  
Naufal Dhiya Ulhaq  
Sharira Saniane

## Daftar Isi

Dasar Teori.....	4
Instalasi.....	4
Instalasi Ansible.....	4
Terminologi Dasar.....	5
Provisioning Vagrant dengan Ansible.....	7
Provision Multiple VM dengan Ansible.....	9
SOAL LATIHAN.....	11
Soal Praktikum Modul 2.....	12

## Dasar Teori

Ansible adalah tool yang dapat digunakan untuk mengotomasi proses provisioning dan konfigurasi server. Ketika kamu ingin membuat server, kamu pasti akan melakukan konfigurasi awal, seperti update & menginstall package lain. Bayangkan jika kamu sedang membuat 10 server, apa tidak lelah harus mengkonfigurasi 10 server satu per satu? Nah di sini lah Ansible menjadi sangat berguna, kita hanya perlu membuat instruksi saja dan Ansible dapat menjalankan instruksi itu langsung di 10 server yang sedang kita setup. Satu hal yang menarik dari Ansible adalah ketika Ansible menjalankan suatu instruksi untuk server, Ansible akan memastikan bahwa server sudah dalam keadaan yang sesuai dengan instruksi. Contoh, ketika kita memberikan instruksi kepada Ansible untuk membuat directory, maka mereka akan memastikan bahwa directoy yang diminta sudah ada pada server, bukan hanya menjalankan command mkdir saja. Selain itu, Ansible juga dikatakan sebagai agentless dimana kamu tidak perlu menginstall Ansible pada setiap server. Kamu bisa menginstall Ansible dalam satu komputer saja dan terhubung dan menjalankan tasks ke server via SSH saja.

## Instalasi

### Instalasi Ansible

Ansible dapat diinstall pada perangkat dengan OS UNIX-based yang mencakup RedHat, Debian, Ubuntu, macOS, BSD, Windows Subsystem for Linux (WSL) dengan requirement sudah terinstall Python versi 3.9 atau yang lebih baru. Ansible tidak dapat diinstall pada Windows secara langsung tanpa WSL. Dengan demikian, disarankan untuk tidak menggunakan Windows sebagai OS perangkat utama yang akan mengontrol seluruh task Ansible.

#### 1. Cara Installasi dengan pip

Untuk melakukan installasi Ansible secara umum untuk semua OS yang disupport, ikuti langkah-langkah berikut.

- a. Pastikan versi Python sudah sesuai (Versi yang didukung adalah Python 3.9 atau yang lebih baru)

```
python --version
```

- b. Pastikan pip sudah terinstall

```
python3 -m pip -V
```

NOTE: Jika pip belum terinstall, silakan ikuti langkah installasi pada dokumentasi berikut.

- c. Instal Ansible

```
python3 -m pip install --user ansible
```

- d. Pastikan Ansible telah terinstall

```
ansible --version
```

## 2. Cara Instalasi Lain untuk Ubuntu or Debian

Untuk melakukan instalasi Ansible pada Linux Ubuntu atau Debian-based, ikuti langkah-langkah berikut.

### a. Lakukan update

```
sudo apt-get update
```

### b. Install dependency untuk Ansible

```
sudo apt-get install software-properties-common
```

### c. Masukkan repository Ansible

```
sudo add-apt-repository --yes --update ppa:ansible/ansible
```

### d. Install Ansible

```
sudo apt-get install ansible
```

### e. Pastikan Ansible telah terinstall

```
ansible --version
```

## NOTE

- Ansible versi terbaru saat ini mendukung Ubuntu 18.04, 20.04, dan yang lebih baru.
- Apabila menggunakan versi Ubuntu yang lebih lama, ubah software-properties-common menjadi python-software-properties.

## 3. Cara Instalasi untuk WSL

Instalasi untuk WSL dapat mengikuti langkah-langkah pada dokumentasi [berikut](#).

# Terminologi Dasar

## 1. Modules

Module adalah sebuah kode yang dapat mengeksekusi command. Ansible menyediakan library module yang dapat dieksekusi secara langsung pada host remote atau melalui playbook.

Contoh Modules

```
ansible localhost -m command -a "/bin/echo 'Hello Ansible'"
```

Keterangan:

- ansible localhost: menjalankan module ini pada komputer kita.
- -m command: menjalankan module "command", module paling simple di Ansible yang berfungsi untuk menjalankan command pada server.
- -a "/bin/echo 'Hello Ansible'": memasukan argument ke module.

## 2. Inventory

Ansible mengotomasi task pada managed hosts menggunakan sebuah list atau kumpulan/grup dari list yang dinamakan dengan inventory. Inventory yang paling simple adalah inventory yang berisi list dari host atau grup. File inventory host umumnya berada pada path `/etc/ansible/hosts`. Kita bisa membuat inventory dalam dua format yang umum yaitu `.ini` dan `.yaml`. Berikut contoh inventory basic.

```
ungrouped:
  hosts:
    mail.example.com:
webservers:
  hosts:
    foo.example.com:
    bar.example.com:
dbservers:
  hosts:
    one.example.com:
    two.example.com:
    three.example.com:
```

### Default Groups

Meskipun kita tidak membuat grup apa pun di file inventory, Ansible akan membuat dua grup default: `all` dan `ungrouped`. Grup `all` berisi setiap host. Grup `ungrouped` berisi semua host yang tidak memiliki grup lain selain `all`. Setiap host akan selalu tergabung dalam setidaknya 2 grup (`all` dan `ungrouped` atau `all` dan beberapa grup lainnya). Misalnya, dalam inventory dasar di atas, host `mail.example.com` termasuk dalam grup `all` dan grup `ungrouped`; host `two.example.com` termasuk dalam grup `all` dan grup `dbservers`. Meskipun `all` dan `ungrouped` selalu ada, keduanya bisa tidak muncul dalam daftar grup.

### Host di Multiple Groups

Seperti yang sebelumnya dijelaskan, kita dapat memasukkan host ke dalam lebih dari satu grup. Kita bisa saja membuat grup yang membagi host menjadi beberapa hal berikut.

- What: Aplikasi, stack, atau layanan (ex: `dbserver`, `webserver`, dll)
- Where: Region, wilayah, storage (ex: `surabaya`, `jakarta`, `medan`, dll)
- When: Tahapan development, untuk menghindari testing pada server production misalnya (ex: `prod`, `test`, dll)

Berikut contohnya dalam file `.yaml`.

```
ungrouped:
  hosts:
    mail.example.com:
webservers:
  hosts:
    foo.example.com:
    bar.example.com:
```

```
dbservers:
  hosts:
    one.example.com:
    two.example.com:
    three.example.com:
east:
  hosts:
    foo.example.com:
    one.example.com:
    two.example.com:
west:
  hosts:
    bar.example.com:
    three.example.com:
prod:
  hosts:
    foo.example.com:
    one.example.com:
    two.example.com:
test:
  hosts:
    bar.example.com:
    three.example.com:
```

### 3. Playbook

Playbooks adalah file yang digunakan oleh Ansible untuk mendeploy dan mengkonfigurasi node pada sebuah inventory. Jika kita ingin mengeksekusi perintah lebih dari satu kali, kita dapat membuat playbook. Playbook dapat digunakan untuk:

- Membuat konfigurasi
- Mengorkestrasi langkah-langkah pada proses manual di banyak machine dengan urutan yang ditentukan.
- Mengeksekusi task secara synchronus atau asynchronus.

## Provisioning Vagrant dengan Ansible

Kita dapat menggunakan Ansible untuk melakukan provision terhadap server kita. Pada modul ini kita akan menggunakan Vagrant yang sudah dijelaskan pada modul 1. Pada contoh di sini kita akan coba untuk install Nginx melalui Ansible pada Vagrant VM. Langkah-Langkah Provision.

#### 1. Cek Vagrant

Pertama, cek terlebih dahulu apakah kamu sudah menginstall Vagrant.

```
vagrant -v
```

#### 2. Buat folder baru, dapat dinamakan testprov.

```
mkdir testprov
```

### 3. Membuat Vagrantfile

Selanjutnya, kita akan menginisiasi vagrant sehingga Vagrantfile dapat dibuat.

```
vagrant init ubuntu/focal64
```

Setelah command di atas dijalankan, maka akan terbuat Vagrantfile secara otomatis.

NOTE:

- ubuntu/focal64 adalah versi ubuntu 20.04.6 LTS.

### 4. Buka Vagrantfile dan edit seperti di bawah ini.

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/focal64"

  config.vm.provider "virtualbox" do |vb|
    vb.memory = "2048"
    vb.cpus = "2"
  end

  config.vm.provision "ansible" do |ansible|
    ansible.playbook = "playbook.yaml"
  end

end
```

### 5. Kemudian, kita buat playbook.

Setelah selesai konfigurasi Vagrantfile, kita akan membuat ansible playbook dengan perintah berikut.

```
nano playbook.yaml
```

Kemudian, kita edit file tersebut sehingga menjadi seperti ini.

```
---
- hosts: all
  tasks:
    - name: Install Nginx to VM
      become: true
      apt:
        name: nginx
        state: latest
        update_cache: true
```

NOTE:

- become diset true agar ansible mendapatkan privilege superuser atau root.
- state adalah status dari aplikasi. Di atas kita set latest untuk mendapatkan aplikasi dengan versi paling baru.
- update\_cache adalah modul yang memberikan ansible instruksi untuk menjalankan perintah yang ekuivalen dengan sudo apt update sebelum menginstall nginx.



6. Kemudian jalankan vagrant

```
vagrant up
```

7. Untuk memverifikasi bahwa playbook telah berjalan. Kita bisa melakukan ssh ke Vagrant yang sudah dibuat.

```
vagrant ssh
```

8. Setelah itu kita bisa menjalankan perintah berikut

```
nginx -v
```

9. Apabila terdapat output seperti di bawah ini berarti ansible sudah menjalankan tugasnya untuk menginstall nginx.

```
nginx version: nginx/1.18.0 (ubuntu)
```

#### NOTE:

- Jika kalian mengubah file playbook, misalnya menambahkan task baru. Kalian perlu menjalankan perintah *vagrant provision* untuk memberitahu vagrant untuk melakukan provisioning kembali.

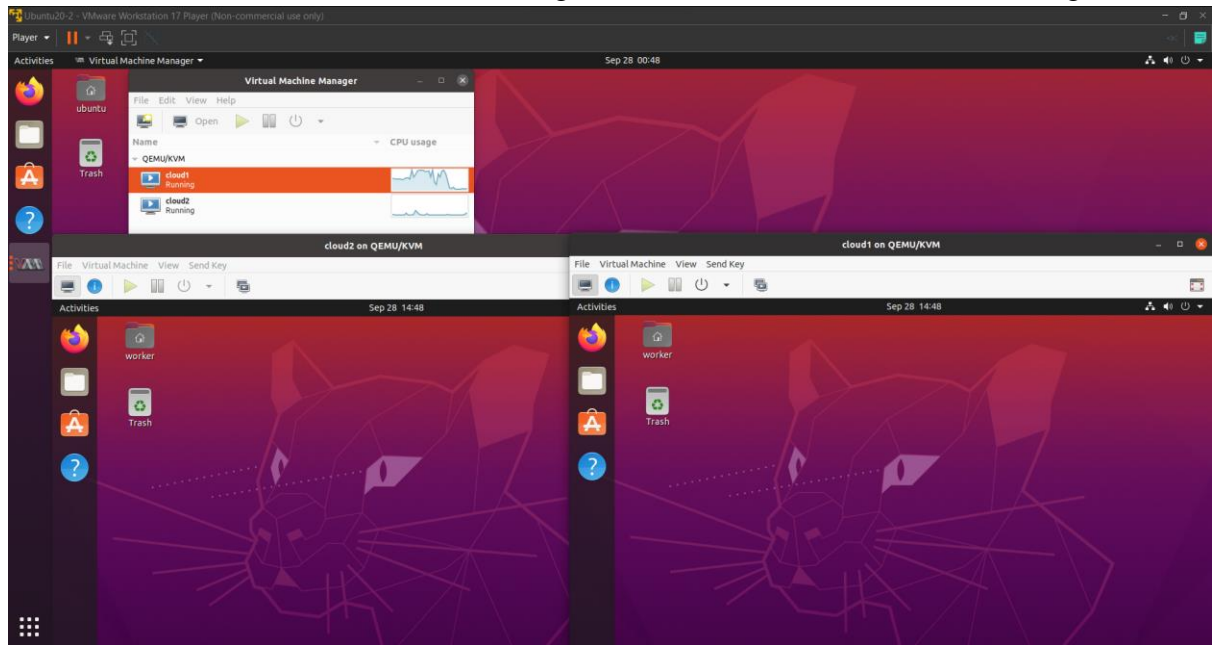
## Provision Multiple VM dengan Ansible

Kita dapat menggunakan Ansible untuk melakukan provision terhadap multiple server yang kita buat. Pada contoh di modul ini, kami menggunakan Ubuntu 20.04.6 LTS sebagai mesin utama dan nantinya pada Ubuntu tersebut akan dibuat virtualisasi dengan KVM (Kernel-based Virtual Machine) yang merupakan virtualisasi native di Linux. Dengan KVM, kita akan membuat dua virtual machine dengan OS Ubuntu.

Di sini kita akan menggunakan 3 mesin, yaitu mesin Ubuntu utama dan 2 virtual machine Ubuntu. Mesin Ubuntu utama akan menjadi host Ansible atau Control Node sedangkan 2 VM akan digunakan sebagai worker yang akan diprovisioning. Langkah-Langkah Provision.

1. Membuat Virtual Machine

Pertama, kita buat dua virtual machine dengan bantuan dari Virtual Machine Manager.



**NOTE:** Untuk installasi KVM dan Virtual Machine Manager dapat dicari di internet ya.

2. Install Ansible di Control Node

Pada Control Node (Ubuntu utama) kita install Ansible terlebih dahulu sesuai intruksi pada materi Installasi.

3. Membuat New Directory

Kemudian, kita buat folder baru untuk testing, misalnya dibuat dengan nama test-playbook.

```
mkdir test-playbook
```

4. Membuat Ansible Inventory

Kemudian, masuk ke folder test-playbook dan buat file inventory ansible yang dinamakan hosts dan diisi dengan.

```
[worker]
worker1    ansible_host=[IP    VM    1]    ansible_ssh_user=worker
ansible_become_pass=worker
worker2    ansible_host=[IP    VM    2]    ansible_ssh_user=worker
ansible_become_pass=worker
```

5. Tes SSH di Worker

Kemudian, coba lakukan ping ke VM worker untuk mengetahui apakah ssh dapat terhubung atau tidak.

```
ansible -i ./hosts -m ping all -k
```

Kemudian, masukkan password login ssh, yaitu worker

**KETERANGAN:**

- Parameter -i untuk mendeclare ansible inventory
- Parameter -m untuk mendeclare module command
- Parameter -k untuk menanyakan password login ssh

- Parameter all untuk penanda ansible dijalankan pada semua host. all dapat diganti juga dengan nama host sesuai keinginan.

**NOTE:**

- Apabila ada error Unreachable, pada tiap worker perlu diinstall openssh-server dan aktifkan port 22 pada setiap worker.
- Apabila ada error Please add host's fingerprint ..., silakan untuk coba koneksi ssh manual terlebih dahulu untuk tiap worker dengan ssh worker@IP untuk menyimpan fingerprintnya pada machine utama.

6. Install Package di Worker

Kemudian, kita membuat playbook sederhana dengan nama install-package.yaml untuk menginstall lynx pada worker

```
- hosts: worker
  tasks:
    - name: Install Lynx
      become: true
      apt:
        name: lynx
        state: latest
        update_cache: true
```

Kemudian, jalankan perintah berikut ini untuk menjalankan playbook.

```
ansible-playbook -i hosts install-package.yaml -k
```

7. Buat Directory dan File Baru di Worker

Terakhir, kita coba untuk membuat folder baru di kedua VM worker. File playbook untuk ini kita namakan newfile.yaml.

```
- hosts: worker
  tasks:
    - name: Create new directory
      command: mkdir -p newdir
    - name: Create new yaml file
      command: touch newfile.yaml
```

Kemudian, jalankan perintah berikut ini untuk menjalankan playbook.

```
ansible-playbook -i hosts newfile.yaml -k
```

## SOAL LATIHAN

Lakukanlah provision Vagrant dengan Ansible untuk menginstall beberapa aplikasi berikut.

- Nginx
- PHP 8.0
- Composer
- Git.

## Soal Praktikum Modul 2

### Ansible

1. Waktu pengerjaan praktikum sesuai dengan timeline yang diberikan (Selasa jam 10.00 WIB sampai Sabtu jam 22.00 WIB).
2. Hasil pengerjaan praktikum dalam bentuk video demonstrasi diupload ke Youtube dengan status unlisted dan format judul TKA\_[Kelompok]\_[Judul Modul] (Contoh: TKA\_E03\_Ansible)
3. Praktikan tidak diperbolehkan menanyakan jawaban dari soal yang diberikan kepada asisten maupun praktikan dari kelompok lainnya.
4. Pengerjaan soal sesuai dengan modul yang telah diajarkan. **Provisioning dan konfigurasi harus menggunakan Ansible.**
5. Jika ditemukan indikasi kecurangan dalam bentuk apapun di pengerjaan soal shift, maka nilai dianggap 0.
6. Pengumpulan video demonstrasi akan dikumpulkan melalui Google Form pada link yang akan dibagikan 1 jam sebelum deadline pengerjaan soal
7. Harap mengumpulkan tepat waktu karena apabila lebih dari waktu yang ditentukan akan mendapat pengurangan poin.
8. **KARENA ALASAN RESOURCE VM YANG DIPERLUKAN SANGAT BESAR UNTUK MENYELESAIKAN SOAL 3 SEBELUMNYA, MAKA SOAL NO 3 KAMI DIGANTI/REVISI TOTAL, MOHON UNTUK MEMPERHATIKAN SOAL TERBARU.**